# Enhancing Road Safety with AI-Driven Traffic Accident Analysis and Prediction

**Student Name:** [B THRISHA]

**Register Number**: [623123104050]

**Institution**: [VSA GROUP OF INSTITUTIONS]

**Department**: COMPUTER SCIENCE AND ENGINEERING

**Date of Submission**:

**1. Problem Statement**

Traffic accidents are a major global concern, leading to thousands of fatalities and injuries every

year. Rapid urbanization, increased vehicle usage, and varying driving conditions have intensified

the risks. Traditional traffic analysis methods fall short in proactively identifying high-risk scenarios.

Hence, there is a pressing need for intelligent systems that can analyze accident patterns and

predict potential accidents, enabling timely preventive measures

2.Objectives of the Project

- To develop an AI-based system that analyzes historical traffic accident data.

- To identify accident-prone zones and conditions.

- To predict future traffic accident trends based on real-time and historical data.

- To offer actionable insights to enhance road safety and inform policymaking

4. Data Sources

- Public datasets from Kaggle, UCI Machine Learning Repository.

- Real-time traffic APIs (e.g., OpenTraffic, Google Maps).

- Weather data from open APIs (e.g., OpenWeatherMap).

- Dataset type: A

mix of static (downloaded) and dynamic (API-based) data.

5. High-Level Methodology

Data Collection:

- Download open datasets from Kaggle and UCI.

- Access real-time traffic and weather data via public APIs.

- Generate synthetic data to balance underrepresented categories.

Data Cleaning:

- Remove duplicate entries and handle missing values using imputation.

- Standardize formats (e.g., date-time, locations).

- Normalize numerical features and encode categorical data.

Exploratory Data Analysis (EDA):

- Use histograms, bar charts, heatmaps, and correlation plots.

- Perform trend analysis by timuting factors to accidents.

Feature Engineering:

- Extract time features like weekday, peak hour, and season.

- Calculate traffic density and weather severity indicators.

- Engineer spatial features such as proximity to intersections or highways.

Model Building:

- Experiment with Random Forest, Gradient Boosting, Neural Networks, and LSTM for time-series

predictions.

- Select models based on interpretability and performance.

Model Evaluation:

- Metrics: Accuracy, precision, recall, F1-score, and ROC-AUC.

- Validation strategy: k-fold cross-validation and test split evaluation.

Visualization & Interpretation:

- Use dashboards built with Plotly or Tableau to present results.

- Generate heatmaps for accident hotspots.

- Provide interpretive summaries and visual storytelling.

Deployment:

- Deploy as a web dashboard using Streamlit or Flask.

- Build REST APIs for model predictions.

- Use Docker for containerization if scaling is needed.

6. Tools and Technologies

- Programming Language: Python

- Notebook/IDE: Google Colab, Jupyter Notebook

- Libraries: Pandas, NumPy, Seaborn, Matplotlib, Scikit-learn, TensorFlow, XGBoost, Plotly

- Optional Tools for Deployment: Streamlit, Flask, FastAPI, Docker

7. Team Members and Roles

THRISHA- Project Lead: Oversees overall project progress, integrates modules.

- POUNRAJ- Data Engineer: Handles data collection, cleaning, and preprocessing.

-PRABAKARAN - Data Scientist: Develops and evaluates machine learning models.

-RAJKUMAR - Visualization Expert: Builds dashboards and data presentations.

.